Automatic Evaluation of Linguistic Validity in Japanese CCG Treebanks

Asa Tomita[1] **Hitomi Yanaka**[2,3] **Daisuke Bekki**[1]
[1]Ochanomizu University [2]The University of Tokyo [3]RIKEN
{tomita.asa, bekki}@is.ocha.ac.jp hyanaka@is.s.u-tokyo.ac.jp

Abstract

In Natural Language Inference, the accuracy of systems based on compositional semantics depends on the quality of syntactic analysis, which in turn relies on linguistically valid training and evaluation data, typically provided by treebanks. However, conventional treebank evaluation metrics focus on data coverage and fail to assess the linguistic validity of syntactic structures. This paper proposes novel evaluation methods to enable automatic and multifaceted assessment of linguistic validity. We apply these methods to a Japanese treebank based on Combinatory Categorial Grammar and report the evaluation results.

1 Introduction

Natural Language Inference (NLI) is one of the core tasks in natural language processing. Among the various approaches to NLI, research on inference systems based on compositional semantics has experienced steady research progress (Mineshima et al., 2015; Abzianidze and Bos, 2017; Hu et al., 2020). In such systems, obtaining linguistically valid syntactic structures and semantic representations is essential because the inference accuracy is strongly influenced by the outputs of syntactic and semantic analyses, which serve as preprocessing for inference. In particular, syntactic and semantic analyses that contain errors can lead to incorrect inference results. Specifically, Japanese syntactic parsers based on Combinatory Categorial Grammar (CCG) (Steedman, 1996, 2000) achieve high accuracy on standard evaluation datasets (Yoshikawa et al., 2017), although their outputs have also been reported to lack linguistic validity, especially when handling complex constructions such as passives and causatives (Bekki and Yanaka, 2023). This discrepancy arises from the fact that both training and evaluation are conducted on treebanks that contain linguistically invalid analyses.

This issue fundamentally stems from the absence of established approaches to evaluating the linguistic validity of treebanks. While many treebanks have been developed for various languages and formal grammars (Marcus et al., 1993; Hockenmaier and Steedman, 2007; Bos et al., 2009; Boxwell and Brew, 2010; Hockenmaier, 2006), their linguistic validity is yet to be evaluated sufficiently. In many cases, the validity of these resources relies on the assumption that they have been constructed or verified by linguists, but such manual assurances do not offer a quantitative measure of linguistic validity. Instead, there is a growing need for principled methods that can evaluate the linguistic validity of treebanks in a systematic way.

Therefore, this study proposes methods for evaluating the linguistic validity of treebanks from both syntactic and semantic perspectives. We apply these evaluation methods to the Japanese CCG treebank constructed by Tomita et al. (2024), and we report our evaluation results.

2 Construction of the CCG treebank

2.1 Combinatory Categorial Grammar

In syntactic parsing grounded in compositional semantics, sentences are transformed into syntactic structures based on formal grammars. Among these, CCG is characterized by having the weakest generative power among mildly context-sensitive grammars in the Chomsky hierarchy, which makes it particularly well-suited for providing sufficient expressivity to capture the essential syntactic structures of sentences. Treebanks based on CCG (Hockenmaier and Steedman, 2007; Tran and Miyao, 2022) are constructed via automatic conversion from treebanks based on Context Free Grammar (CFG; Marcus et al., 1993) or dependency structures (Nivre et al., 2020), and they are used as training and evaluation data for existing syntactic parsers (Yoshikawa et al., 2017; Tian et al., 2020).

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 74–80
August 28-29, 2025 ©2025 Association for Computational Linguistics

2.2 Linguistically Valid Japanese CCG Treebank

A representative CCG treebank for Japanese is the Japanese CCGbank (Uematsu et al., 2013), which was constructed via automatic conversion from a corpus of dependency structures. It has been widely used as training and evaluation data for Japanese CCG parsers. However, Bekki and Yanaka (2023) pointed out that the Japanese CCGbank contains errors in the analysis of sentences involving case alternations such as passive and causative constructions. To address this issue, Tomita et al. (2024) proposed a new method called *Reforging* (described in Section 2.2.2) that uses the Japanese syntactic parser *lightblue* (Bekki and Kawazoe, 2016) to construct *lightblue CCGbank*, a Japanese CCG treebank designed specifically with a focus on linguistic validity.

2.2.1 Japanese Syntactic Parser *lightblue*

lightblue is a Japanese syntactic parser based on CCG. Its lexicon follows the theoretical analysis of Bekki (2010), particularly in the design of syntactic features, allowing it to generate structures with detailed information such as verb conjugation forms. In contrast, the Japanese CCGbank lacks such feature granularity, making it difficult to constrain syntactic structures—especially for closed-class words.

For open-class words, *lightblue* partially builds its lexicon using lexical information from the morphological analyzer Juman (Kawahara and Kurohashi, 2006). Unlike neural parsers trained on treebanks, it parses without supervision, relying on a precompiled lexicon and CCG combinatory rules. However, inaccuracies in predicate-argument structures remain, limiting the linguistic validity of its analyses.

2.2.2 Overview of Reforging

To address the issue of *lightblue* mentioned above, prior work (Tomita et al., 2024) has introduced a module that integrates argument structures into lexical entries by extracting them from external linguistic resources (Kubota et al., 2020; Ueda et al., 2023) and modifies *lightblue* lexical entries by adding or removing argument structure information. This module in combination with the parser *lightblue* forms the treebank method called *Reforging*. Using this method, the *lightblue CCGbank* was constructed as a linguistically valid Japanese CCG treebank. The dataset consists of 13,653 sentences extracted from ABCTreebank (Kubota et al., 2020), each assigned a CCG syntactic structure and semantic representation based on Dependent Type Semantics (DTS; Bekki and Mineshima (2017)). The remaining challenge is how to evaluate the linguistic validity of the *lightblue CCGbank*.

3 Treebank Evaluation

3.1 Conventional Evaluation Metrics

Treebanks are typically evaluated using metrics such as lexical coverage and parser accuracy. However, in this section, we point out that these conventional evaluation metrics are not comprehensive and are insufficient for evaluating the linguistic validity of treebanks.

3.1.1 Lexical Entries and Coverage Rate

A lexicon can be constructed from the words appearing at the leaf nodes of a parse tree, and metrics such as the number of lexical entries and lexical coverage can be used to evaluate the comprehensiveness of the treebank. Lexical coverage refers to the proportion of words for which the grammar assigns a gold-standard category.

It is important to note that high lexical coverage does not guarantee the linguistic validity of the dataset. Coverage merely indicates how extensively the lexicon can assign some category to encountered words, but it does not evaluate whether the treebank data itself is linguistically valid. Therefore, even a high coverage rate does not ensure the quality or validity of the data.

3.1.2 Parsing Accuracy

In parser-based evaluation, a treebank is used to train the parser, and its accuracy is evaluated by measuring how well it can analyze the syntactic structures of input sentences. Software tools such as evalb[1] are commonly used to compute metrics including precision, recall, F-score, and tagging accuracy.

Although parsing accuracy is commonly used to evaluate syntactic parsers, it does not necessarily reflect the linguistic validity of the underlying dataset. Since accuracy measures alignment with gold-standard annotations, a parser may achieve high scores even when trained on erroneous data. Accordingly, high parsing accuracy alone cannot be taken as evidence of a linguistically valid treebank.

[1] https://nlp.cs.nyu.edu/evalb/

3.2 Evaluation of Linguistic Validity

As discussed in Section 3.1, conventional methods for evaluating treebanks are not sufficient for the quantitative evaluation of linguistic validity. Moreover, evaluating CCG syntactic structures requires advanced knowledge of computational linguistics, making manual evaluation costly and impractical for large-scale treebank validation.

Therefore, this study proposes an automatic method for evaluating the linguistic validity of large-scale Japanese CCG treebanks. In *lightblue CCGbank*, each sentence is assigned a CCG syntactic structure and DTS semantic representation. Building on this data, we introduce two evaluation metrics, one for syntax and one for semantics. By combining these metrics, a multidimensional evaluation approach is achieved.

3.2.1 Syntax-Based Evaluation

Because all sentences in *lightblue CCGbank* are extracted from ABCTreebank, each syntactic structure in the former corresponds to one in the latter. Assuming that ABCTreebank, which was constructed via expert annotation, provides linguistically valid structures, we evaluate the reliability of *lightblue CCGbank* by scoring its alignment with ABCTreebank.

The ABC grammar used in ABCTreebank is a form of categorial grammar that employs function application and composition rules. However, because the definitions of syntactic categories and unary rules differ between ABC grammar and CCG, direct comparison is impossible. To enable comparison, the syntactic categories in ABCTreebank are converted to their CCG counterparts, and alignment is scored based on the following procedure as shown in Figure 1:

1. Convert the ABC grammar into CCG.
2. For each syntactic structure obtained in 1 and its counterpart in *lightblue CCGbank*, create a list of pairs consisting of syntactic categories and phonetic forms.
3. Calculate the score as the proportion of elements in ABCTreebank list that are included in the *lightblue CCGbank* list.

This method has two advantages: one is to compare empty categories in CCG with unary rules in ABCTreebank, and another is to accommodate differences in predicate analysis. However, it also has limitations: it assumes that ABCTreebank is entirely correct, which may not necessarily be the case, and it cannot evaluate syntactic features not annotated in ABCTreebank.

3.2.2 Semantics-Based Evaluation

All syntactic structures in *lightblue CCGbank* are assigned DTS semantic representations. DTS is a proof-theoretic semantic framework based on Dependent Type Theory (DTT; Martin-Löf (1984)). We propose a method for evaluating the validity of DTS semantic representations using type-theoretic verification, known as “type checking”. Type checking is a procedure for verifying whether a semantic representation has a well-formed type; if the representation can be proven to have the type `type`, then the check is considered successful.

Type checking fails when the semantic representation is ill-formed. However, it is theoretically proven that when CCG and DTS are used as the syntactic and semantic frameworks, semantic representations should always be well-typed (Bekki, Forthcoming). Therefore, a failure in type checking suggests errors in the implementation of lexical items or combinatory rules that yield ill-typed semantic representations cannot be considered linguistically valid under this system. This property enables the evaluation of syntactic validity from the perspective of semantic compositionality.

A notable strength of this method is that it evaluates syntactic structures at the semantic level based on type theory. However, passing type checking does not necessarily imply linguistic validity of the associated syntactic structures. Thus, syntactic scores and type-theoretic verification serve complementary functions, and their combined use is essential for a comprehensive assessment of treebank quality.

4 Evaluation Experiment

4.1 Experimental Setup

In total, 760 sentences were sampled from various genres within *lightblue CCGbank* and used for evaluation. The syntactic structures were comprehensively evaluated based on the following metrics.

Syntactic Structure Score Average Using the method in Section 3.2.1, each sentence was scored by the percentage of matching (`surface form`, `syntactic category`) pairs, and averages were calculated per genre.

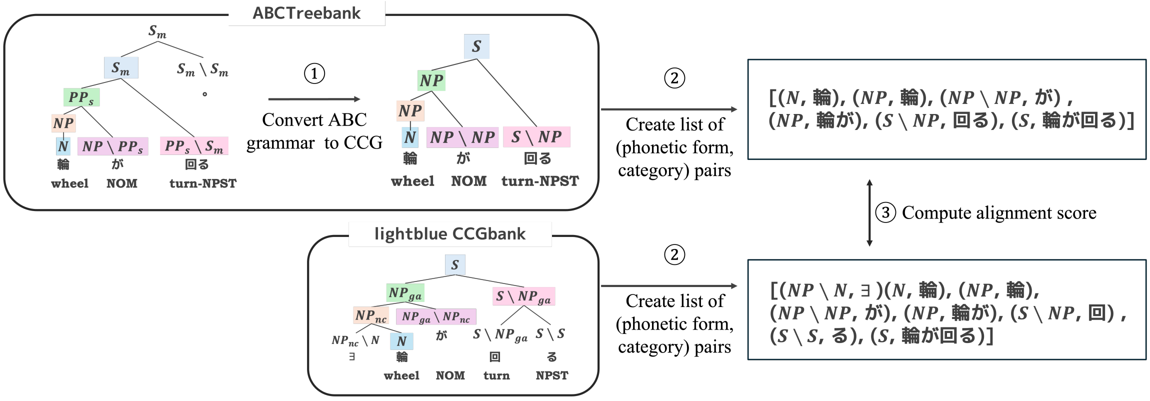
ABCTreebank
S_m
S_m
$S_m \backslash S_m$
。
PP_s
NP
N
$NP \backslash PP_s$
$PP_s \backslash S_m$
輪
が
回る
wheel
NOM
turn-NPST
① Convert ABC grammar to CCG
S
NP
NP
N
$NP \backslash NP$
$S \backslash NP$
輪
が
回る
wheel
NOM
turn-NPST
② Create list of (phonetic form, category) pairs
[(N, 輪), (NP, 輪), (NP \ NP, が) , (NP, 輪が), (S \ NP, 回る), (S, 輪が回る)]
③ Compute alignment score
lightblue CCGbank
S
NP_{ga}
$S \backslash NP_{ga}$
NP_{nc}
$NP_{ga} \backslash NP_{nc}$
$NP_{nc} \backslash N$
N
$S \backslash NP_{ga}$
$S \backslash S$
∃
輪
が
回
る
wheel
NOM
turn
NPST
② Create list of (phonetic form, category) pairs
[(NP \ N, ∃)(N, 輪), (NP, 輪), (NP \ NP, が), (NP, 輪が), (S \ NP, 回) , (S \ S, る), (S, 輪が回る)]

Figure 1: Scoring Process in Syntactic Evaluation

Genre	Number of Data	Average Score	Type Checking Pass Rate	Overall Score
Aozora Bunko	125	42.4	63.2	20.89
Bible	40	49.1	57.5	21.57
Books	10	49.8	60.0	33.33
Dictionary	100	55.59	57.0	28.57
Proceedings	35	41.8	77.1	23.91
Fiction	30	51.1	66.7	31.82
Law	10	33.4	80.0	28.57
Other	50	50.2	64.0	26.47
News	50	40.4	78.0	21.88
Non-fiction	10	53.4	100.0	33.33
Spoken Language	50	36.98	88.0	25.37
TED Talks	25	41.68	64.0	21.88
Textbooks	200	49.59	60.0	27.54
Wikipedia	25	45.88	88.0	32.43
Total	760	46.0	66.2	26.00

Table 1: Evaluation Results

Type Checking Passage Rate Based on Section 3.2.2, type checking was performed on the semantic representations. The passage rate was defined as the proportion of sentences with semantic representations successfully verified as well-typed.

Overall Evaluation This metric is the percentage of sentences with a syntactic score of 50.0 or higher that also passed type checking, indicating both syntactic and semantic validity.

4.2 Results and Discussion

The results of the experiment are presented in Table 1. The overall average syntactic structure score was 46.0, with 503 out of the 760 evaluated sentences passing type checking, yielding a pass rate of 66.2%.

An important finding is that there is no clear correlation between the average syntactic structure score average and the type checking passage rate, suggesting that the two metrics capture orthogonal properties relevant to linguistic validity. For instance, in the law genre, although the type checking passage rate is as high as 88%, the syntactic score remains relatively low at 33.4%, indicating that syntactic alignment and semantic well-formedness are independent. This observation highlights the complementary nature of the two evaluation metrics. Even if a parse tree receives a high syntactic score, indicating structural similarity to gold-standard annotations, it cannot be regarded as linguistically valid if it fails type checking. Passing the type checking procedure serves as a necessary condition for semantic consistency, verifying that the semantic composition is correctly implemented within the DTS framework.

4.3 Manual Evaluation

To assess the reliability of our syntax-based evaluation metric, we compared its results against a manually annotated subset of 152 sentences from the lightblue CCGbank. The results are shown in Table 2.

The metric achieved a precision of 0.64, recall of 0.79, F1-score of 0.71, and accuracy of 0.74 with respect to human judgments. These results suggest that the syntax-based evaluation has relatively high recall, meaning it is capable of capturing most linguistically valid structures identified by human annotators. However, the lower precision indicates that some sentences deemed valid by the metric may not align with human judgments, possibly due to overpermissive category matching. Overall, the moderate F1-score (0.71) and reasonably high accuracy (0.74) indicate that the syntax-based metric can serve as a useful proxy for linguistic validity, though it may require further refinement to reduce false positives.

		Manual Evaluation	
		True	False
Score > 50	True	48	27
	False	13	64
Accuracy		0.739	
Precision		0.640	
Recall		0.787	
F1		0.706	

Table 2: Confusion Matrix and Manual Evaluation Results

5 Limitations and Future Work

5.1 Annotation Errors in the ABCTreebank

Although the average syntactic alignment score appears relatively low at 46%, this result is partially attributable to annotation errors in the ABCTreebank, which serves as the gold standard in our evaluation. Our evaluation assumes that ABCTreebank provides linguistically valid structures; hence, any inaccuracies in its annotations directly affect the computed scores.

For instance, determiners are annotated as N/N, a category that yields a noun. However, they should more appropriately be labeled as NP/N, since they functionally yield noun phrases. Such inconsistencies in category assignment can reduce alignment scores, even when the underlying syntactic structures are otherwise linguistically sound.

5.2 Limits of Cross-Framework Evaluation

Some category mismatches observed in our evaluation — such as annotating determiners as N/N in ABCTreebank, while they are assigned NP/N in lightblue CCGbank — might appear to be minor inconsistencies. However, such differences are not simply attributable to the annotation rules; rather, they reflect deeper theoretical assumptions about the treatment of syntactic categories. In CCG, for example, NP/N indicates that a determiner produces a complete noun phrase, aligning with its semantic interpretation and compositional properties. In contrast, frameworks like ABC grammar often avoid using NP entirely, opting for a more uniform treatment of nouns and noun phrases.

This highlights a broader challenge for our evaluation method; it is not simply a conversion from one formal description to another, but a translation between distinct linguistic theories. Consequently, it necessitates a careful alignment of theoretical assumptions across frameworks. Each theory prioritizes different linguistic principles. Without explicitly addressing these theoretical discrepancies, evaluation scores may primarily reflect inter-framework divergences rather than actual linguistic inaccuracies. In other words, a mismatch between NP/N and N/N might not indicate a parsing error, but rather a fundamental theoretical difference in how the grammar encodes syntactic categories.

5.3 Future Work

While our evaluation is currently conducted within the CCG and DTS frameworks, the proposed metrics are designed to be framework-agnostic. Future work will involve investigating their applicability to other syntactic and semantic frameworks, such as CFG and Abstract Meaning Representations (Langkilde and Knight, 1998), thereby further substantiating the generality of our evaluation method. Moreover, we intend to enhance the validity of the *lightblue CCGbank* through the incorporation of feedback mechanisms into the treebank construction process.

6 Conclusion

This study proposed syntactic and semantic evaluation metrics for assessing the linguistic validity of treebanks from two independent perspectives. These metrics enable a more fine-grained analysis of structural validity than conventional approaches. Ensuring the validity of treebanks is essential not only for improving inference accuracy but also for satisfying requirements such as transparency of error detection and enhanced explainability in future language processing systems. By addressing the lack of principled evaluation methods for linguistic validity, this work offers a step toward more reliable and linguistically grounded approaches in NLP.

Acknowledgments

This work was supported by JST BOOST, Japan Grant Number JPMJBS2406, JSPS KAKENHI Grant Number JP23H03452, Japan, and JST CREST Grant Number JPMJCR20D2, Japan.

References

Lasha Abzianidze and Johan Bos. 2017. Towards universal semantic tagging. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS) — Short papers*.

Daisuke Bekki. 2010. *Nihongo-Bunpoo-no Keisiki-Riron - Katuyootaikei, Toogohantyuu, Imigoosei - (trans. 'Formal Japanese Grammar: the conjugation system, categorial syntax, and compositional semantics')*. Kuroshio Publisher, Tokyo.

Daisuke Bekki. Forthcoming. From Dependent Type Theory to natural language semantics.

Daisuke Bekki and Ai Kawazoe. 2016. Implementing variable vectors in a CCG parser. In *Logical Aspects of Computational Linguistics. Celebrating 20 Years of LACL (1996–2016)*, pages 52–67, Berlin, Heidelberg. Springer Berlin Heidelberg.

Daisuke Bekki and Koji Mineshima. 2017. *Context-Passing and Underspecification in Dependent Type Semantics*, pages 11–41. Springer International Publishing, Cham.

Daisuke Bekki and Hitomi Yanaka. 2023. Is Japanese CCGBank empirically correct? a case study of passive and causative constructions. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 32–36, Washington, D.C. Association for Computational Linguistics.

Johan Bos, Cristina Bosco, and Alessandro Mazzei. 2009. Converting a dependency treebank to a categorial grammar treebank for Italian. In *Proceedings of the eighth international workshop on treebanks and linguistictheories (TLT8)*, pages 27–38, Italy, Milan.

Stephen A. Boxwell and Chris Brew. 2010. A pilot Arabic CCGbank. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Julia Hockenmaier. 2006. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 505–512, Sydney, Australia. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kuebler. 2020. MonaLog: a lightweight system for natural language inference based on monotonicity. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 334–344, New York, New York. Association for Computational Linguistics.

Daisuke Kawahara and Sadao Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 1344–1347, Genoa, Italy. European Language Resources Association (ELRA).

Yusuke Kubota, Koji Mineshima, Noritsugu Hayashi, and Shinya Okano. 2020. Development of a general-purpose categorial grammar treebank. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5195–5201, Marseille, France. European Language Resources Association.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 704–710, Montreal, Quebec, Canada. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Per Martin-Löf. 1984. *Intuitionistic Type Theory Vol. 1*. Bibliopolis.

Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061, Lisbon, Portugal. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Mark Steedman. 1996. *Surface Structure and Interpretation*. The MIT Press, Cambridge.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

Yuanhe Tian, Yan Song, and Fei Xia. 2020. Supertagging Combinatory Categorial Grammar with attentive graph convolutional networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6037–6044, Online. Association for Computational Linguistics.

Asa Tomita, Hitomi Yanaka, and Daisuke Bekki. 2024. Reforging : A method for constructing a linguistically valid Japanese CCG treebank. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 196–207, St. Julian's, Malta. Association for Computational Linguistics.

Tu-Anh Tran and Yusuke Miyao. 2022. Development of a multilingual CCG treebank via Universal Dependencies conversion. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5220–5233, Marseille, France. European Language Resources Association.

Nobuhiro Ueda, Kazumasa Omura, Takashi Kodama, Hirokazu Kiyomaru, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. 2023. KWJA: A unified Japanese analyzer based on foundation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 538–548, Toronto, Canada.

Sumire Uematsu, Takuya Matsuzaki, Hiroki Hanaoka, Yusuke Miyao, and Hideki Mima. 2013. Integrating multiple dependency corpora for inducing wide-coverage Japanese CCG resources. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1042–1051, Sofia, Bulgaria. Association for Computational Linguistics.

Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* CCG parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 277–287, Vancouver, Canada. Association for Computational Linguistics.